CIND830
Python Programming for Data Science
Lab 6 - Answer Key

1 Functions

Create a file called `funLab6Q1.py` then add the following code lines to it.

```
def iSum(listNum):
    '''' Return the sum of all the numbers inside the list.
    >>> iSum([10,20,30])
    60
    >>> iSum([1.5, 2.5, 5.5])
    9.5 '''
    return (totalSum)
iString = input('Enter a list of numbers separated by space: ')
iList = iString.split()
print("The total sum is", iSum(iList))
```

a. Run the code inside the `funLab6Q1.py` file, then document the error that will be displayed by the interpreter.

```
---------------------------------------------------------------------------
NameError                 Traceback (most recent call last)
<ipython-input-16-1d6ce93abc5c> in <module>
     11 iString = input('Enter a list of numbers separated by space: ')
     12 iList = iString.split()
---> 13 print("The total sum is", iSum(iList))
     14
<ipython-input-16-1d6ce93abc5c> in iSum(listNum)
      7     9.5
      8     '''
----> 9     return(totalSum)
     10
     11 iString = input('Enter a list of numbers separated by space: ')
NameError: name 'totalSum' is not defined
---------------------------------------------------------------------------
```

b. Fix the error displayed in **1a** by adding the necessary commands to compute the `totalSum` variable, then display it on the screen upon calling the `iSum` function.

```
def iSum(listNum):
    ''' (number, number) -> number
    Return the sum of all the numbers inside the list.
    >>> iSum([10,20,30])
    60
    >>> iSum([1.5, 2.5, 5.5])
    9.5
    '''
    totalSum = 0
    for iPos in range(0, len(listNum)):
        totalSum += float(listNum[iPos])
    return(totalSum)

iString = input('Enter a list of numbers separated by space: ')
iList = iString.split()
print("The total sum is", iSum(iList))
```

c. Add to the `funLab6Q1.py` file another function called `iAverage` to compute the arithmetic average of the entered numbers, then display the results on the screen.

```
def iSum(listNum):
    ''' (number, number) -> number
    Return the sum of all the numbers inside the list.
    >>> iSum([10,20,30])
    60
    >>> iSum([1.5, 2.5, 5.5])
    9.5
    '''
    totalSum = 0
    for iPos in range(0, len(listNum)):
        totalSum += float(listNum[iPos])
    return(totalSum)

def iAverage(listNum):
    ''' (number, number) -> number
    Return the sum of all the numbers inside the list.
    >>> iAverage([10,20,30])
    20
    >>> iAverage([1.5, 2.5, 5.5])
    3.17
```

```
    '''
    return(iSum(iList)/len(listNum))


iString = input('Enter a list of numbers separated by space: ')
iList = iString.split()
print("The total sum is\tt", iSum(iList))
print("The arithmetic average is\t", iAverage(iList))
```

d. Review the Python documentation provided for the `statistics` module and how to call statistical functions to complete the following tasks. The mathematical statistics functions can be found at the following link:
https://docs.python.org/release/3.7.0/library/statistics.html

i. From the `statistics` module, load the following functions: `mean()`, `median()`, `mode()`, and `stdev()`
Hint: Use the `import` statement to bind the module or function name into the local scope.

```
from statistics import mean, median, mode, stdev.
```

ii. Compute the `mean()`, `median()`, and `mode()` of the numbers inside a given list.

```
from statistics import mean, median, mode, stdev
iString = input('Enter a list of numbers separated by space: ')
iList = [float(i) for i in iString.split()]
print("The arithmetic average is\t", mean(iList))
print("The 50th percentile is\t", median(iList))
print("The most occured number is\t", mode(iList))
```

iii. Display the description of the `harmonic_mean()`, then use this function after importing it to calculate the harmonic average of the numbers in a given list. Hint: Use the built-in `help()` function.

```
from statistics import mean, harmonic_mean
iString = input('Enter a list of numbers separated by space: ')
iList = [float(i) for i in iString.split()]
print("The arithmetic average is\t", mean(iList))
print("The harmonic average is\t", harmonic_mean(iList))
```

2 Classes

Consider the following code:

```
ceStudent = Student('Harry', 'IND-830', 3.9)
print(ceSudent.nameStd)
'Harry'
print(ceStudent.subjCode)
'IND-830'
print(ceStudent.gpaGrade)
3.9
```

a. Try to run the first line of the aforementioned code and write down the error message that will be generated by the interpreter.

```
'NameError: name 'Student' is not defined'
```

b. Create a class called `Student` with a constructor method `__init__` that has 4 arguments: Student, the student's name, subject code and the respective GPA grade. Then run the code at the header of the question and observe the results.

```
class Student:
    ''' A student in Ryerson U. '''
    def __init__(self, nameStd, subjCode, gpaGrade):
        ''' (Student, str, str, str)
        Create a new student named nameStd, with subjCode and gpaGrade.
        '''
        self.nameStd = nameStd
        self.subjCode = subjCode
        self.gpaGrade = gpaGrade
```

```
ceStudent = Student('Harry', 'IND-830', 3.9)
print(ceStudent.nameStd) #Harry
print(ceStudent.subjCode) # IND-830
print(ceStudent.gpaGrade) #3.9
```

c. In class `Student`, declare a method called `is_diffName` that takes two `Student` objects and returns `True` if the name of the first `Student` object does not match the name of the second.

```
class Student:
    ''' A student at Ryerson U. '''
    def __init__(self, nameStd, subjCode, gpaGrade):
        ''' (Student, str, str, str)
        Create a new student named nameStd, with subjCode and gpaGrade
        '''
        self.nameStd = nameStd
        self.subjCode = subjCode
        self.gpaGrade = gpaGrade
    def is_diffName(self, std2):
        if self.nameStd != std2.nameStd:
            return(True)
        else:
            return(False)
```

d. Verify the correctness of the `is_diffName` method by running the following code.

```
ceStudent1 = Student('Harry', 'IND-830', 3.9)
ceStudent2 = Student('Harry', 'IND-830', 3.9)
ceStudent3 = Student('Sally', 'IND-830', 3.9)

ceStudent1.is_diffName(ceStudent2) #Should Return False
ceStudent1.is_diffName(ceStudent3) #Should Return True
```

e. Create a super class called `UnivMember` with 4 arguments: UnivMember, the member's name, address and email. Then modify the `Student` class to inherit all the attributes of the `UnivMember` class.

```
class UnivMember:
    ''' A University member'''
    def __init__(self, nameMem, addressMem, emailMem):
        self.nameMem = nameMem
        self.addressMem = addressMem
        self.emailMem = emailMem

class Student(UnivMember):
    ''' A student at Ryerson U. '''
    def __init__(self, nameMem, addressMem, emailMem\
                 , subjCode, gpaGrade):
        super().__init__(nameMem, addressMem, emailMem)
        self.subjCode = subjCode
        self.gpaGrade = gpaGrade
```

f. Verify the correctness of the `UnivMember` class by using the following code.

```
ceStudent1 = Student('Harry', 'Montreal'\
                     , 'harry@ryerson.ca', 'IND-830', 3.9)
ceStudent2 = Student('Sally', 'Toronto'\
                     , 'sally@ryerson.ca', 'IND-830', 4.1)

print(ceStudent1.emailMem)  # Should display harry@ryerson.ca
print(ceStudent1.gpaGrade)  # Should display 3.9
```

This is the end of lab 6
Tamer Abdou, PhD